JPL SuperMOCA Demonstration Testbed

Carlos Carrión

(carlos.carrion@jpl.nasa.gov)

Jet Propulsion Laboratory

SCWG

June 3, 1997

Agenda

- Purpose of JPL Demos
- Current JPL Demo
    - Scenario Description
    - Notes
    - Configuration
- Updates to Current JPL Demo
    - Scenario Description
    - Notes
    - Configuration
- Planned JPL Demo Development
    - Scenario Description
   - Other Work in Progress
- Summary
- Next Steps

Purpose of JPL Demos

- Short Term
    - What is the set of SMS[1] services needed for space missions
    - What does it mean/take to build virtual devices for space missions
    - Viability of virtual devices for space missions
    - Applicability of device descriptions to space missions
    - Demonstrate the SuperMOCA technology areas : SMS and Information Architecture
- Long Term
    - What implications do virtual devices have on S/C design, i.e.:
        - Communications
        - Data System Architecture
        - Processing resources: computing power and memory
    - Plug 'n Play
    - What modifications to MMS[2] services are needed to do space missions
    - What additional MMS services are needed for space missions
    - Which MMS services are not needed for space missions
    - Mission Operations interoperability
    - Gain enough information to generate SuperMOCA specifications

1: Space Messaging System   2: Manufacturing Message Specification

Carlos Carrión   JPL   3

Current JPL Demonstration
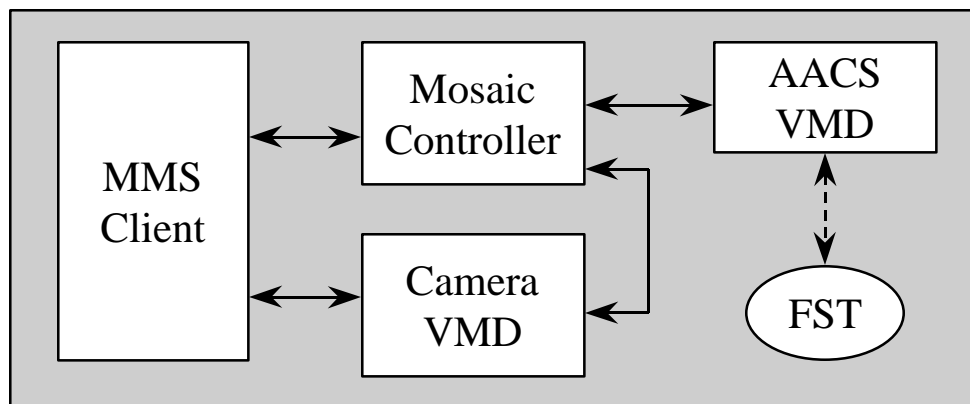
Current JPL Demo

Scenario Description

Objects:
- Client
- Spacecraft:
    - Mosaic Controller
    - Attitude & Articulation Control Subsystem
    - Camera Simulator

User wants to do an mxn mosaic centered on some target:

1. User uses the Client object to specify:
    - Number of rows, columns, overlap in x/y direction, target quaternion.
2. User can tell Camera Simulator object to compress the "images".
3. User uses the Client object to connect to the Mosaic Controller and Camera.
4. Client object starts the mosaic.
5. Mosaic Controller object controls the mosaic execution.
6. Mosaic Controller object notifies the Client object when the mosaic starts/ends.
7. User can download the "images" from the Camera Simulator.
8. User can tell Camera object to delete the "images" from the Camera file store.

Current JPL Demo (cont.)

Notes

- Camera is a simulation based on the FST[1] camera simulator flight software:
  - Images are just strings of bytes for now.
- AACS[2] object is used as the interface to the AACS running in the FST:
- 13 MMS services used:
  - 2 Context Management
  - 2 VMD[3] Support
  - 3 Variable Access & Management
  - 3 File Access & Management
  - 3 Event & Alarm Management

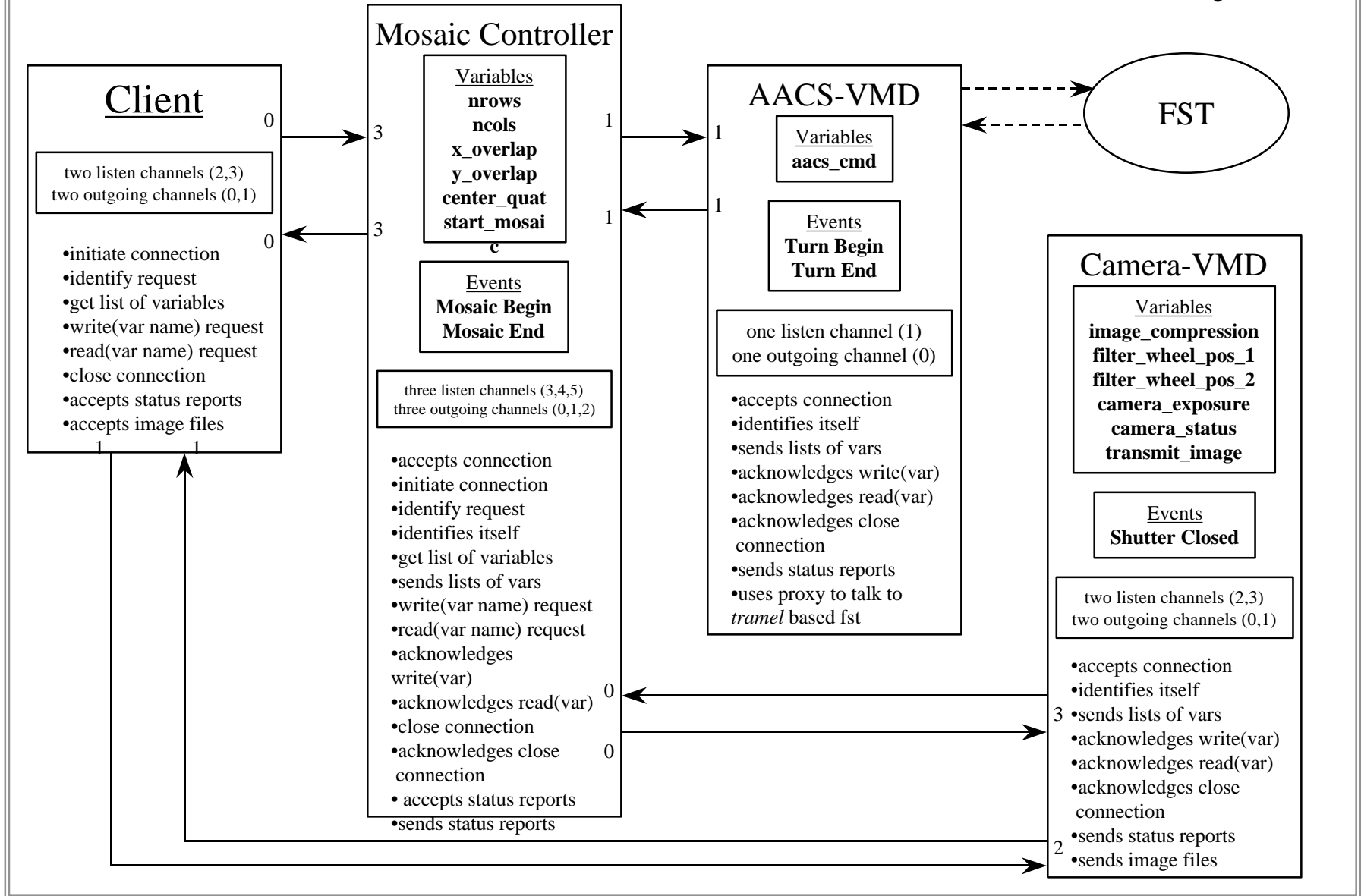1: JPL Flight System Testbed   2: Attitude & Articulation Control Subsystem   3: Virtual Manufacturing Device (or Virtual Device)
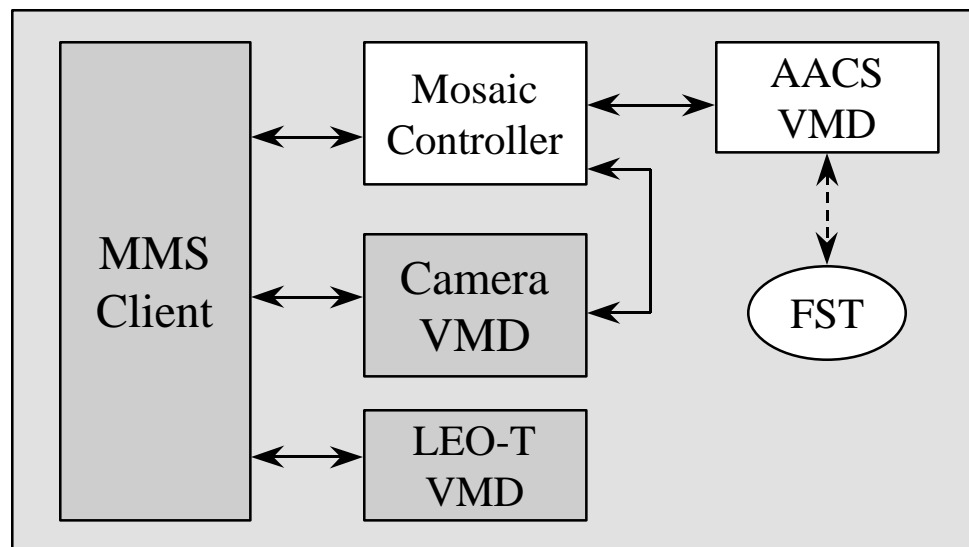   see backup slides for a definition

Current JPL Demo (cont.)

MMS Client - MMS Mosaic Controller -MMS AACS VMD - MMS Camera VMD Configuration

Mosaic Controller

Variables
nrows
ncols
x_overlap
y_overlap
center_quat
start_mosaic

Events
Mosaic Begin
Mosaic End

three listen channels (3,4,5)
three outgoing channels (0,1,2)

•accepts connection
•initiate connection
•identify request
•identifies itself
•get list of variables
•sends lists of vars
•write(var name) request
•read(var name) request
•acknowledges write(var)
•acknowledges read(var)
•close connection
•acknowledges close connection
• accepts status reports
•sends status reports

Client

two listen channels (2,3)
two outgoing channels (0,1)

•initiate connection
•identify request
•get list of variables
•write(var name) request
•read(var name) request
•close connection
•accepts status reports
•accepts image files

AACS-VMD

Variables
aacs_cmd

Events
Turn Begin
Turn End

one listen channel (1)
one outgoing channel (0)

•accepts connection
•identifies itself
•sends lists of vars
•acknowledges write(var)
•acknowledges read(var)
•acknowledges close connection
•sends status reports
•uses proxy to talk to *tramel* based fst

FST

Camera-VMD

Variables
image_compression
filter_wheel_pos_1
filter_wheel_pos_2
camera_exposure
camera_status
transmit_image

Events
Shutter Closed

two listen channels (2,3)
two outgoing channels (0,1)

•accepts connection
•identifies itself
•sends lists of vars
•acknowledges write(var)
•acknowledges read(var)
•acknowledges close connection
•sends status reports
•sends image files

Updates to Current JPL Demo

Updates to Current JPL Demo
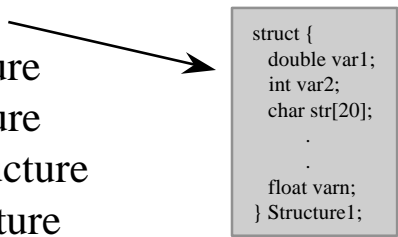
Scenario Description

Objects:

- **Client**
- Spacecraft:
    - Mosaic Controller
    - Attitude & Articulation Control Subsystem
    - **Camera Simulator**
- **Ground Terminal**
    - **LEO-T[1] Simulator**

- Same scenario as current demo with the following additions:
    - Capability to monitor (not yet control) a LEO-T station from the same MMS based client used to communicate with the S/C.
    - User able to instruct Camera Simulator to execute maintenance scripts and report results to the Client.
    - The User also is able to uplink a modified maintenance script to the Camera Simulator.

1: Low Earth Orbit Tracking Station

Updates to Current JPL Demo (cont.)

Notes

- The Program Invocation Services and Domain Management Services were used to implement the maintenance script capability. Domains are writeable.
- The LEO-T VMD is not controllable. It sends InfoReports every 60 seconds containing the following complex types:

```
struct {
   double var1;
   int var2;
   char str[20];
      .
      .
   float varn;
} Structure1;
```

  - Antenna Data Structure
  - Framing Data Structure
  - Scheduling Data Structure
  - Pass Info Type Structure
    - This demonstrates the cability of downlinking complex types to the Client.
- 18 MMS services used.
  - 2 Context Management
  - 2 VMD Support
  - 3 Variable Access & Management
  - 3 File Access & Management
  - 3 Event & Alarm Management
  - 3 Domain Management
  - 2 Program Invocation

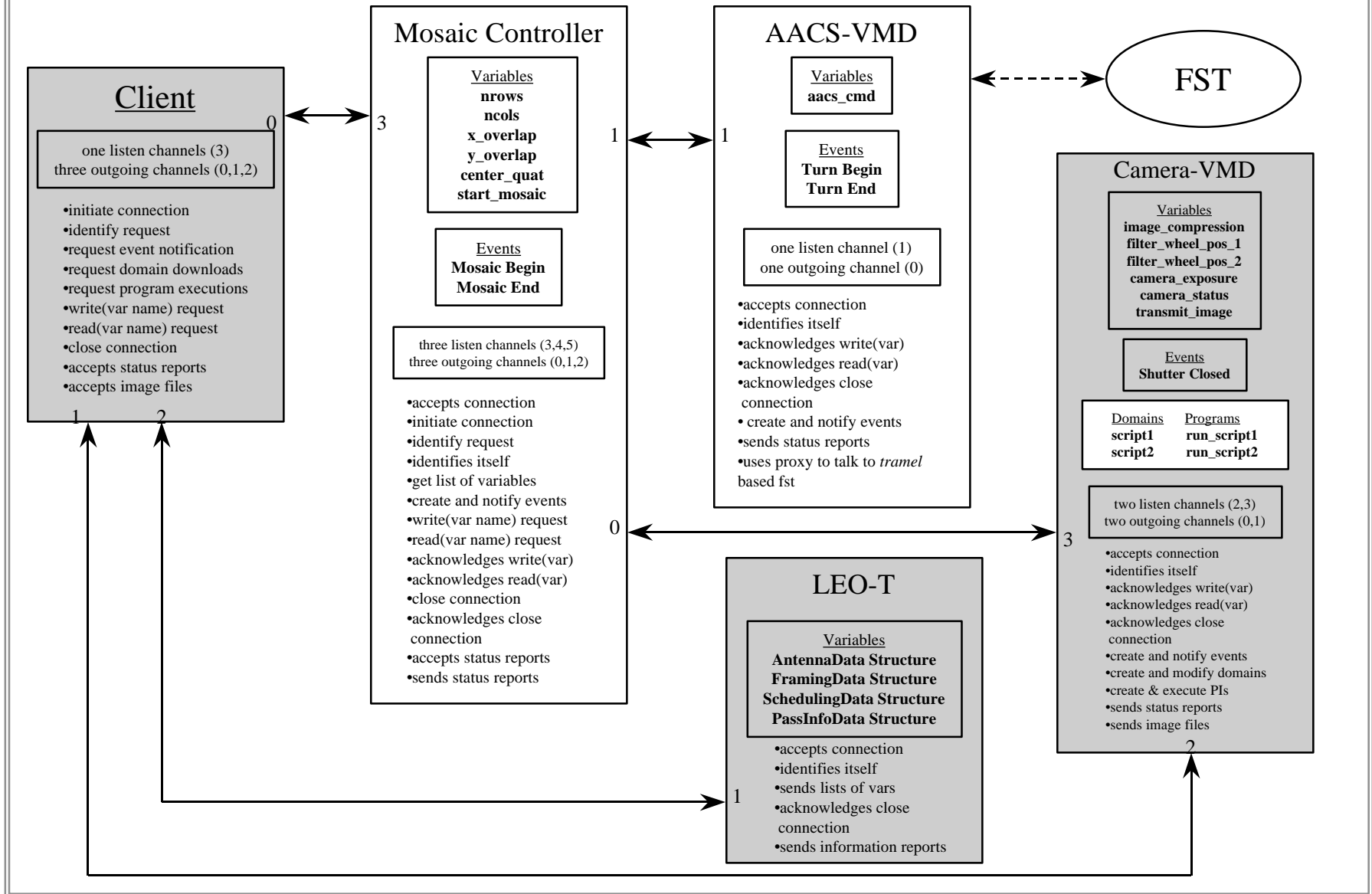- Porting Demonstration to the JPL-SMOCC[1]. Currently running on SunOS based SUN SPARC.

    1: Simulated Mission Operations Control Center

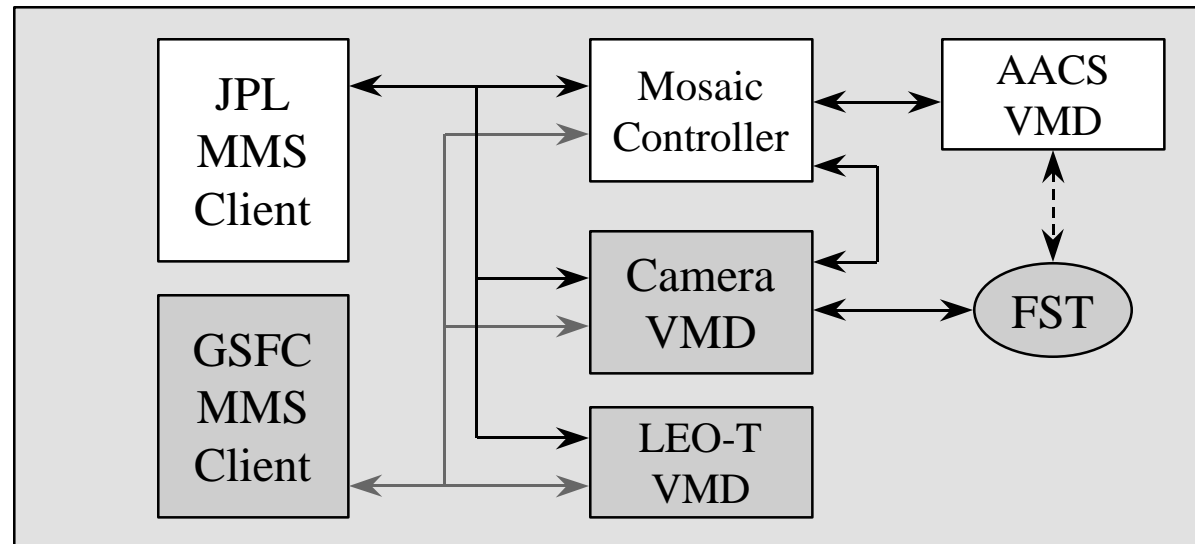Carlos Carrión   10

Updates to Current JPL Demo (cont.)

MMS Client - MMS Mosaic Controller -MMS AACS VMD - MMS Camera VMD - MMS LEO-T VMD Configuration

Client

0

one listen channels (3)
three outgoing channels (0,1,2)

•initiate connection
•identify request
•request event notification
•request domain downloads
•request program executions
•write(var name) request
•read(var name) request
•close connection
•accepts status reports
•accepts image files

1   2

Mosaic Controller

3   1

Variables
nrows
ncols
x_overlap
y_overlap
center_quat
start_mosaic

Events
Mosaic Begin
Mosaic End

three listen channels (3,4,5)
three outgoing channels (0,1,2)

•accepts connection
•initiate connection
•identify request
•identifies itself
•get list of variables
•create and notify events
•write(var name) request
•read(var name) request
•acknowledges write(var)
•acknowledges read(var)
•close connection
•acknowledges close
 connection
•accepts status reports
•sends status reports

0

AACS-VMD

1

Variables
aacs_cmd

Events
Turn Begin
Turn End

one listen channel (1)
one outgoing channel (0)

•accepts connection
•identifies itself
•acknowledges write(var)
•acknowledges read(var)
•acknowledges close
 connection
• create and notify events
•sends status reports
•uses proxy to talk to *tramel*
based fst

FST

Camera-VMD

Variables
image_compression
filter_wheel_pos_1
filter_wheel_pos_2
camera_exposure
camera_status
transmit_image

Events
Shutter Closed

Domains   Programs
script1   run_script1
script2   run_script2

two listen channels (2,3)
two outgoing channels (0,1)

3

•accepts connection
•identifies itself
•acknowledges write(var)
•acknowledges read(var)
•acknowledges close
 connection
•create and notify events
•create and modify domains
•create & execute PIs
•sends status reports
•sends image files

2

LEO-T

Variables
AntennaData Structure
FramingData Structure
SchedulingData Structure
PassInfoData Structure

1

•accepts connection
•identifies itself
•sends lists of vars
•acknowledges close
 connection
•sends information reports

Carlos Carrión

JPL

11

Planned JPL Demo Development

Planned JPL Demo Development

Scenario Description

Objects:

- **Client (JPL and GSFC)**
- Spacecraft:
    - Mosaic Controller
    - Attitude & Articulation Control Subsystem
    - **Camera Simulator**
- **Low Earth Orbit Tracking Station**

- Same scenario as previous demo with the following additions:
    - Modify Camera VMD to allow images (created by the FST scene generator) to be returned by the Camera VMD to the Client.
    - Implement Camera simulator device description language "database" in the demo. This will allow "automatic" configuration of the Camera VMD based on the contents of the device description.
    - GSFC will be able to monitor and control the JPL demo remotely.

Planned JPL Demo Development (cont.)

- Goddard Space Flight Center work in progress:
  - Windows based MMS client object running with Labview GUI.
  - GUI/DDL[1] Interface
  - Implementing LEO-T data types.
  - Will demonstrate interoperability: Different NASA centers can monitor and control the JPL demo and configure their Client GUI from DDL information.
  - MMS Connectivity between GSFC and JPL-SMOCC achieved 2 weeks ago.

- LEO-T simulator will be replaced with the actual station application software.

1: Device Description Language

Carlos Carrión   14

Summary

- **Current JPL demo**
    - Rudimentary mosaic capability using 13 MMS services.
- **Updates to JPL demo**
    - Additional camera simulator capabilities implemented.
    - LEO-T simulator added.
    - 18 MMS services being used.
- **Planned development**
    - Add the ability to return real images from camera simulator.
    - Add device description capability.
    - Replace LEO-T simulator with "real" thing.
    - Implement interface between GUI/DDL.
    - Continue GSFC work to demonstrate interoperability.

Next Steps

- Add more devices to the Demo
    - S/W simulation
        - FST simulated devices
    - Real devices
        - Possible cooperation with X2000
- Proposal for LEO-T control.